

Fast, Linear Time, m-Adic Hierarchical Clustering for Search and Retrieval using the Baire Metric, with linkages to Generalized Ultrametrics, Hashing, Formal Concept Analysis, and Precision of Data Measurement

Fionn Murtagh (1,3) and Pedro Contreras (2,3)

(1) Science Foundation Ireland, Wilton Park House, Wilton Place, Dublin 2, Ireland

(2) Thinking Safe Ltd., Royal Holloway University of London, Egham TW20 0EX, England

(3) Department of Computer Science, Royal Holloway

University of London, Egham TW20 0EX, England

fmurtagh@acm.org, pedro.contreras@thinkingsafe.com

We describe many vantage points on the Baire metric and its use in clustering data, or its use in preprocessing and structuring data in order to support search and retrieval operations. In some cases, we proceed directly to clusters and do not directly determine the distances. We show how a hierarchical clustering can be read directly from one pass through the data. We offer insights also on practical implications of precision of data measurement. As a mechanism for treating multidimensional data, including very high dimensional data, we use random projections.

I. INTRODUCTION

In areas such as search, matching, retrieval and general data analysis, massive increase in data requires new methods that can cope well with the explosion in volume and dimensionality of the available data. In this work, the Baire metric, which is furthermore an ultrametric, is used to induce a hierarchy and in turn to support clustering, matching and other operations.

Arising directly out of the Baire distance is an ultrametric tree, which also can be seen as a tree that hierarchically clusters data. This presents a number of advantages when storing and retrieving data. When the data source is in numerical form this ultrametric tree can be used as an index structure making matching and search, and thus retrieval, much easier.

The clusters can be associated with hash keys, that is to say, the cluster members can be mapped onto “bins” or “buckets”.

Another vantage point in this work is precision of measurement. Data measurement precision can be either used as given or modified in order to enhance the inherent ultrametric and hence hierarchical properties of the data.

Rather than mapping pairwise relationships onto the reals, as distance does, we can alternatively map onto subsets of the power set of, say, attributes of our observation set. This is expressed by the generalized ultrametric, which maps pairwise relationships into a partially ordered set. It is also current practice as formal concept analysis where the range of the mapping is a lattice.

Relative to other algorithms the Baire-based hierarchical clustering method is fast. It is a direct reading algorithm involving one scan of the input data set, and is of linear computational complexity.

Many vantage points are possible, all in the Baire metric framework. The following vantage points will be discussed in this article.

- Metric that is simultaneously an ultrametric.
- Hierarchy induced through m-adic encoding (m positive integer, e.g. 10).
- p-Adic (p prime) or m-adic clustering.
- Hashing of data into bins.
- Data precision of measurement implies how hierarchical the data is.
- Generalized ultrametric.
- Lattice-based formal concept analysis.
- Linear computational time hierarchical clustering.

II. THE BAIRE METRIC, THE BAIRE ULTRAMETRIC

A. Metric and Ultrametric Spaces

Our purpose consists of mapping data into an ultrametric space or, alternatively expressed, searching for an ultrametric embedding, or ultrametrization [42]. Actually, inherent ultrametricity leads to an identical result relative to most commonly used agglomerative criteria [35]. Furthermore, data coding can help greatly in finding how inherently ultrametric data is [36], and this is further discussed in section VI.

A metric space (X, d) consists of a set X on which is defined a *distance function* d which assigns to each pair of points of X a distance between them, and satisfies the following four axioms for any triplet of points x, y, z :

1. $\forall x, y \in X, d(x, y) \geq 0$ (positiveness);
2. $\forall x, y \in X, d(x, y) = 0 \text{ iff } x = y$ (reflexivity);
3. $\forall x, y \in X, d(x, y) = d(y, x)$ (symmetry);
4. $\forall x, y, z \in X, d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality).

An *ultrametric space* respects the “strong triangular inequality” or *ultrametric inequality* defined as:

$$d(x, z) \leq \max \{d(x, y), d(y, z)\},$$

in addition to the positivity, reflexivity and symmetry properties for any triplet of points $x, y, z \in X$.

Various properties of an ultrametric space ensue from this. For example, the triangle formed by any triplet is necessarily isosceles, with the two large sides equal; or is equilateral. Every point of a circle in an ultrametric space is a center of the circle. Two circles of the same radius, that are not disjoint, are overlapping [2, 28]. Additionally, an ultrametric is a distance that is defined strictly on a tree, which is a property that is very useful in classification [4].

B. Ultrametric Baire Space and Distance

A Baire space consists of countably infinite sequences with a metric defined in terms of the longest common prefix: the longer the common prefix, the closer a pair of sequences. What is of interest to us is this longest common prefix metric, which we call the Baire distance [7, 33, 38].

We begin with the longest common prefixes at issue being digits of precision of univariate or scalar values. For example, let us consider two such decimal values, x and y , with both measured to some maximum precision. One or both can be padded with 0s to have this maximum precision. With no loss of generality we take x and y to be bounded by 0 and 1. Thus we consider ordered sets x_k and y_k for $k \in K$. So $k = 1$ is the first decimal place of precision; $k = 2$ is the second decimal place; . . . ; $k = |K|$ is the $|K|$ th decimal place. The cardinality of the set K is the precision with which a number, x or y , is measured.

Consider as examples $x_3 = 0.478$; and $y_3 = 0.472$. Start from the first decimal position. For $k = 1$, we find $x_1 = y_1 = 4$. For $k = 2$, $x_2 = y_2 = 7$. But for $k = 3$, $x_3 \neq y_3$.

We now introduce the following distance (case of vectors x and y , with 1 attribute, hence unidimensional):

$$d_{\mathcal{B}}(x_K, y_K) = \begin{cases} 1 & \text{if } x_1 \neq y_1 \\ \inf \mathcal{B}^{-n} & x_n = y_n, \quad 1 \leq n \leq |K| \end{cases} \quad (1)$$

We call this $d_{\mathcal{B}}$ value Baire distance, which is a 1-bounded ultrametric [7, 38] distance, $0 < d_{\mathcal{B}} \leq 1$. When dealing with binary (boolean) data 2 is the chosen base, $\mathcal{B} = 2$. When working with real numbers the base is best defined to be 10, $\mathcal{B} = 10$. With $\mathcal{B} = 10$, for instance, it can be seen that the Baire distance is embedded in a 10-way tree which leads to a convenient data structure to support search and other operations when we have decimal data. As a consequence data can be organized, stored and accessed very efficiently and effectively in such a tree.

For \mathcal{B} prime, this distance has been studied by Benois-Pineau et al. [3] and by Bradley [6], with many further (topological and number theoretic, leading to algorithmic and computational) insights arising from the p-adic (where p is prime) framework.

III. MULTIDIMENSIONAL USE OF THE BAIRE METRIC THROUGH RANDOM PROJECTIONS

It is well known that traditional clustering methods do not scale well in very high dimensional spaces. A standard and widely used approach when dealing with high dimensionality is to apply a dimensionality reduction technique. This consists of finding a mapping F relating the input data from the space \mathbb{R}^d to a lower-dimension feature space \mathbb{R}^k : $F : \mathbb{R}^d \rightarrow \mathbb{R}^k$.

A least squares optimal way of reducing dimensionality is to project the data onto a lower dimensional orthogonal subspace. Principal component analysis (PCA) is a popular choice to do this. It uses a linear transformation to form a simplified (viz. reduced dimensionality), dataset while retaining the characteristics (viz. variances) of the original data. PCA selects a best fitting, ordered sequence of subspaces (of dimensionality 1, 2, 3, ...) that best preserve the variance of the data.

This is a good solution when the data allows these calculations, but PCA as well as other dimensionality reduction techniques remain expensive from a computational point of view, for very large data sets. The essential eigenvalue and eigenvector decomposition is of $O(d^3)$ computational complexity. Looking beyond the least squares and orthogonal PCA projection, we have studied the benefits of random projections.

Random projection [5, 12, 15, 17, 18, 29, 31, 43] is the finding of a low dimensional embedding of a point set, such that the distortion of any pair of points is bounded by a function of the lower

dimensionality.

The theoretical support for random projection can be found in the Johnson-Lindenstrauss Lemma [27]. It states that a set of points in a high dimensional Euclidean space can be projected into a low dimensional Euclidean space such that the distance between any two points changes by a fraction of $1 + \varepsilon$, where $\varepsilon \in (0, 1)$.

Johnson-Lindenstrauss Lemma

For any $0 < \varepsilon < 1$ and any integer n , let k be a positive integer such that

$$k \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln n \quad (2)$$

Then for any set V of any points in \mathbb{R}^d , there is a map $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$ such that for all $u, v \in V$,

$$(1 - \varepsilon) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon) \|u - v\|^2$$

.

Furthermore, this map can be found in randomized polynomial time.

The original proof [27] was further simplified by Frankl and Maehara [19], and Dasgupta and Gupta [13]. See also [1, 43].

In practice we find that random directions of high dimensional vectors are a sufficiently good approximation to an orthogonal system of axes. We present experimental results in [10]. In this way we can exploit data sparsity in high dimensional spaces.

In random projection the original d -dimensional data is projected to a k -dimensional subspace ($k \ll d$), using a random $k \times d$ matrix R :

$$X'_{k \times N} = R_{k \times d} X_{d \times N} \quad (3)$$

where $X_{d \times N}$ is the original set with d -dimensionality and N observations. From the computational aspect, forming the random matrix R and projecting the $d \times N$ data matrix X into the k dimensions is of order $O(dkN)$. If X is sparse with c non-zero entries per column, the complexity is of order $O(ckN)$.

Random projection can be seen as a class of hashing function. (This is further discussed in section V below.) Hashing is much faster than alternative methods because it avoids the pairwise comparisons required for partitioning and classification. This process is depicted in a Euclidean

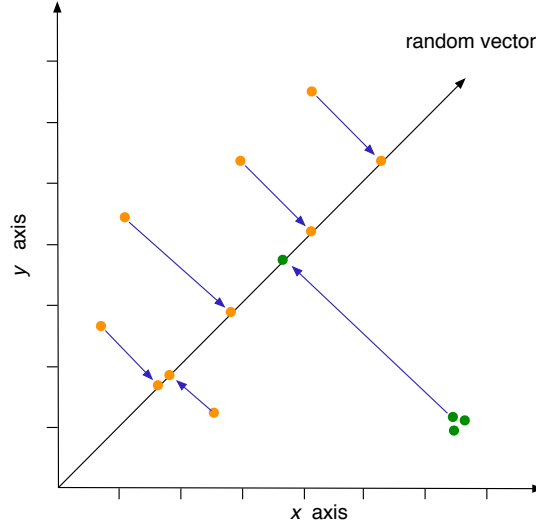


FIG. 1: Random projection axis, showing orthogonal projections.

2-dimensional space in Figure 1, where a random vector is drawn and data points are projected onto it. If two points (p, q) are close, they will have a very small $\|p - q\|$ (Euclidean metric) value; and they will hash to the same value with high probability. If they are distant, they should collide with small probability.

IV. HIERARCHICAL TREE DEFINED FROM M-ADIC ENCODING

Consider data values, base 10, that are $> 0, < 1$. Let the full data set be associated with the root of a regular 10-way tree. Determine 10 clusters/bins at the first level from the root of the tree, labeled through the first digit of precision, 0, 1, 2, ... , 9. Determine the first level of the tree – for each of the 10 first level clusters – labeled through the second digit of precision. The clusters/bins, associated with terminals in the tree, can be labeled 00, 01, 02, ... , 09; 10, 11, ... , 19; 20, 21, ... , 29; 90, 91, 92, ... , 99. This m-adic encoding tree, with $m = 10$, can be continued through further levels.

In [11], a large cosmology survey data set is used, where it is sought to match spectrometric redshifts against photometric redshifts (respectively denoted z_{spec} and z_{phot}). Redshift is (i) cosmological distance, (ii) recession velocity, (iii) look-back time to the observation in question, and (iv) the third dimension of the 3D cosmos in addition to (in the commonly used, extra-Solar System, coordinate frame) right ascension and declination. Using spectrometry is demanding in terms of instrumental procedure, but furnishes better quality output (in terms of precision and

error). Photometric redshifts, on the other hand, are more easily obtained but are of lower quality. Hence there is interest in inferring spectrometric redshifts from photometric redshifts. With that aim in mind, in [11] we looked at a clusterwise regression approach, where “clusterwise” is not spatial but rather based on measurement precision.

To summarize our results on approximately 400,000 pairs of redshift measurements, we found the following.

- 82.8% of z_{spec} and z_{phot} have at least 2 common prefix digits.

This relates to numbers of redshift couples sharing 6, 5, 4, 3, or 2 (precision-ordered) decimal digits.

We can find very efficiently where these 82.8% of the astronomical objects are, in our data.

- 21.7% of z_{spec} and z_{phot} have at least 3 common prefix digits.

This relates to numbers of observations sharing 6, 5, 4, or 3 decimal digits.

This exemplifies how we read off clusters from the hierarchical tree furnished by the Baire (ultra)metric.

V. LONGEST COMMON PREFIX AND HASHING

The longest common prefix, used in the Baire metric, can be viewed as a hashing or data binning scheme. We will follow up first on the implications of this when used in tandem with random projections for multivariate data (i.e., data in a Euclidean or other space of dimensionality > 1).

A. From Random Projection to Hashing

Random projection is the finding of a low dimensional embedding of a point set – dimension equals 1, hence a line or axis, in this work – such that the distortion of any pair of points is bounded by a function of the lower dimensionality [43]. As noted in subsection III, there is extensive literature in this area, e.g. [16]. While random projection *per se* will not guarantee a bijection of best match in original and in lower dimensional spaces, our use of projection here is effectively a hashing method. We aim to deliberately find hash collisions that thereby provide a sufficient condition for the mapped vectors to be matched. Alternatively expressed, candidate best

match vectors are determined in this way. As an example of this approach, Miller et al. [32] use the MD5 (Message Digest 5) hashing scheme as a basis for nearest neighbor searching. Buaba et al. [8] note that hashing is an approximate matching approach whereby (i) the probability of not finding a nearest neighbor is very small, and (ii) neighbors in the same hash class furnished by the hashing projection are good approximations to the (optimal) nearest neighbor. Retrieval of audio data is at issue in [32], retrieval of Earth observation data is studied in [8], and content-based image retrieval is the focus of [44].

A hash function, used for similarity finding, maps data into a fixed length data *key*. Thus, possibly following random projection, assume our data includes two strings of values 0.457891 and 0.457883456. Now consider how both of these can be put into the same “bin” or “bucket” labeled by 4578. In this case the fixed length hash key of a data value is read off as the first 4 significant digits.

Collision of identically valued vectors is guaranteed, but what of collision of non-identically valued vectors, which we want to avoid? Such a result can be established based on the assumption of what distribution our original data follow. A stable distribution is used in [24], viz. a distribution such that a limited number of weighted sums of the variables is also itself of the same distribution. Examples include both Gaussian (which is 2 stable, [23]) and power law (long tailed) distributions.

Interestingly, however, very high dimensional (or equivalently, very low sample size or low N) data sets, by virtue of high relative dimensionality alone, have points mostly lying at the vertices of a regular simplex or polygon [22, 36]. Such regular sparsity is one reason why we have found random projection to work well. Another reason is that we use attribute weighting (e.g. [38]). Li et al. [30] (see their section 5) note that pairwise distance can be “meaningless” when using heavy tailed distributions but this problem is bypassed by attribute weighting which modifies the data’s 2nd and higher order moments. Thereafter, with the random projection mapping using statistically uniformly drawn weights for attribute j , w_j , then the random projections for data vectors x and y are respectively $\sum_j w_j x_j$ and $\sum_j w_j x'_j$. We can anticipate near equal x_j and x'_j terms, for all j , to be mapped onto fairly close resultant scalar values.

We adopted an experimental approach to confirm these hypotheses, viz., that high dimensional data are “regular” or “structured” in such a way; and that, as a consequence, hashing is particularly well-behaved in the sense of non-identical vectors being nearly always collision-free. We studied stability of results, and also effectiveness relative to other clustering methods, in particular k-means partitioning. In [38], this principle of binning data is used on a large, high dimensional chemoinformatics data set. It is shown in [11] how a large astronomical data set also lends itself

very well to similarity finding in this way.

VI. ENHANCING ULTRAMETRICITY THROUGH PRECISION OF MEASUREMENT

By reducing the precision of measurement we are in effect mapping the data into bins, or hashing the data. Furthermore the longest common prefix as used in the Baire metric gives us one way, that is both practical and useful, to extract reduced precision information from our data.

A. Quantifying Ultrametricity

We assume Euclidean space. If necessary our data can be mapped into a Euclidean space, see e.g. [37] that maps contingency table data endowed with the chi squared metric into a Euclidean factor space. In a different application domain where data are given as pairwise comparisons or preferences, multidimensional scaling methods take pairwise ranks (hence ordinal data) and perform a mapping into a Euclidean space.

Consider, in Euclidean space, a triplet of points that defines a triangle. Take the smallest internal angle, a , in triangle ≤ 60 degrees. For the two other internal angles, b and c , if $|b - c| < 2$ degrees then we characterize the triangle as being approximately isosceles with small base, or equilateral. That is to say, we consider 2 degrees to be an arbitrary small angle. Because of the approximation involved we could claim, informally, that this leads to a fuzzy definition of ultrametricity.

Any triangle in Euclidean space is ultrametric if it is isosceles with small base, or equilateral [28].

To use this in practice, we look for the overall proportion of such triangles in our data. This yields a coefficient of ultrametricity [36]. Therefore to quantify ultrametricity we take all possible triplets, i, j, k . We look at their angles, and judge whether or not the ultrametric triangle properties are verified. Having examined all possible triangles, our ultrametricity measure that we term α is the number of ultrametric-verifying triangles, divided by the total number of triangles. When all triangles respect the ultrametric properties this yields $\alpha = 1$; if no triangle does, then $\alpha = 0$. For N objects, exhaustive enumeration of triangles is computationally prohibitive, so we sample i, j, k in practice. We sample uniformly over the object set.

At least two other approaches to quantifying ultrametricity have been used but in [36] we point to their limitations. First, there is the relationship between subdominant ultrametric, and given

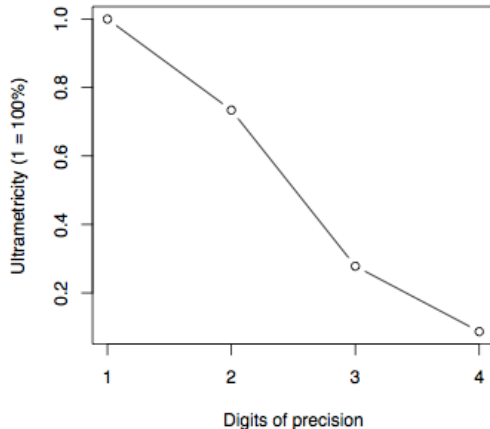


FIG. 2: Dependence of ultrametricity, i.e. data inherently hierarchical, on precision

dissimilarities. See [40]. Secondly, we may look at whether interval between median and maximum rank dissimilarity of every set of triplets is nearly empty. Taking ranks provides scale invariance. This is the approach of [28].

B. Ultrametricity is Pervasive

We find experimentally that ultrametricity is pervasive in the following ways. See [22, 36].

1. As dimensionality increases, so does ultrametricity.
2. In very high dimensional spaces, the ultrametricity approaches being 100%.
3. Relative density is important: high dimensional and spatial sparsity mean the same in this context.

Compared to metric embedding, and model fitting in general, mapping metric (or other) data into an ultrametric, or embedding a metric in an ultrametric, leads to study of distortion.

As such a distortion, let us look at recoding of data, by modifying the data precision. By a focus on the data measurement process, we can find a new way to discover (hierarchical) structure in data.

In Figure 2, 20,000 encoded chemicals were used, normalized as described in [38]. Next, 2000 sampled triangles were selected, and ultrametricities obtained for precisions 1, 2, 3, 4, ... in all

values. Numbers of non-degenerate triangles (out of 2000) were found as follows (where non-degenerate means isosceles with small base):

- precision 1: 2
- precision 2: 1062
- precision 3: 1999
- precision 4: 2000

Thus if we restrict ourselves to just 1 digit of precision we find a very high degree of ultrametricity, based – to be noted – on the preponderance of equilateral triangles. With 2 digits of precision, there are a lot more cases of isosceles triangles with small base.

Thus we can bring about higher ultrametricity in a data set through reducing the precision of data measurement.

VII. GENERALIZED ULTRAMETRIC AND FORMAL CONCEPT ANALYSIS

The usual ultrametric is an ultrametric distance, i.e. for a set I , $d : I \times I \longrightarrow \mathbb{R}^+$. The range is the set of non-negative reals.

The generalized ultrametric is: $d : I \times I \longrightarrow \Gamma$, where Γ is a partially ordered set, or poset. In other words, the *generalized* ultrametric is a set. The range of this generalized ultrametric is therefore defined on the power set or join semilattice. The minimal element of the poset generalizes the 0 distance of the mapping onto \mathbb{R}^+ case.

Comprehensive background on ordered sets and lattices can be found in [14]. A review of generalized distances and ultrametrics can be found in [41]. Generalized ultrametrics are used in logic programming and, as we will discuss in the subsection to follow, formal concept analysis can be seen as use of the generalized ultrametric.

To illustrate how the generalized ultrametric is a vantage point on the Baire framework, we focus on the common, or shared, prefix aspect of this. Consider the following data values: $x_1 = 0.4573, x_2 = 0.4843, x_3 = 0.45635, x_4 = 0.4844, x_5 = 0.4504$. Common prefixes are as follows, where we take decimal digits (i.e. “4573” in the case of x_1).

$$d(x_1, x_2) = 4$$

$$d(x_1, x_3) = 4, 45$$

$$d(x_1, x_4) = 4$$

$$x(x_1, x_5) = 4, 45$$

$$d(x_2, x_3) = 4$$

$$d(x_2, x_4) = 4, 48, 484$$

$$d(x_2, x_5) = 4$$

$$d(x_3, x_4) = 4$$

$$d(x_3, x_5) = 4, 45$$

$$d(x_4, x_5) = 4$$

The partially ordered set is just the structure with the “subset of the common prefix” binary relation with, on one level, the single valued common prefixes (here: 4); the next level has the common prefixes of length 2 (here: 45, 48); the following level has the common prefixes of length 3 (here: 484). Prior to the first level, we have the length 0 common prefix corresponding to no match between the strings.

Thus we see how we can read off a partially ordered set, contained in a lattice, based on the common or shared prefixes.

A. Formal Concept Analysis

In formal concept analysis (FCA) [14], we focus on the lattice as described in the previous subsection. Thus, we can say that a lattice representation is yet another way of displaying (and structuring) the clusters and the cluster assignments in our Baire framework.

Following a formulation by Mel Janowitz (see [25, 26]), lattice-oriented FCA is contrasted with poset-oriented hierarchical clustering in the following way, where by “summarize” is meant that the data structuring through lattice or poset allows statements to be made about the cluster members or other cluster properties.

- Cluster, then summarize. This is the approach taken by (traditional) hierarchical clustering.
- Summarize, then cluster. This is, in brief, the approach taken by FCA.

Further description of FCA is provided in [11]. Our aim here has been to show how the common prefixes of strings leads to the Baire distance, and also to a generalized ultrametric, and furthermore to a poset and an embedding in a lattice.

VIII. LINEAR TIME AND DIRECT READING HIERARCHICAL CLUSTERING

A. Linear Time, or $O(N)$ Computational Complexity, Hierarchical Clustering

A point of departure for our work has been the computational objective of bypassing computationally demanding hierarchical clustering methods (typically quadratic time, or $O(N^2)$ for N input observation vectors), but also having a framework that is of great practical importance in terms of the application domains.

Agglomerative hierarchical clustering algorithms are based on pairwise distances (or dissimilarities) implying computational time that is $O(N^2)$ where N is the number of observations. The implementation required to achieve this is, for most agglomerative criteria, the nearest neighbor chain, together with the reciprocal nearest neighbors, algorithm (furnishing inversion-free hierarchies whenever Bruynooghe’s reducibility property, see [35], is satisfied by the cluster criterion).

This quadratic time requirement is a worst case performance result. It is most often the average time also since the pairwise agglomerative algorithm is applied directly to the data without any preprocessing speed-ups (such as preprocessing that facilitates fast nearest neighbor finding). An example of a linear average time algorithm for (worst case quadratic computational time) agglomerative hierarchical clustering is in [34].

With the Baire-based hierarchical clustering algorithm, we have an algorithm for linear time worst case hierarchical clustering. It can be characterized as a divisive rather than an agglomerative algorithm.

B. Grid-Based Clustering Algorithms

The Baire-based hierarchical clustering algorithm has characteristics that are related to grid-based clustering algorithms, and density-based clustering algorithms, which – often – were developed in order to handle very large data sets.

The main idea here is to use a grid like structure to split the information space, separating the dense grid regions from the less dense ones to form groups. In general, a typical approach within this category will consist of the following steps [21]:

1. Creating a grid structure, i.e. partitioning the data space into a finite number of non-overlapping cells.
2. Calculating the cell density for each cell.

3. Sorting of the cells according to their densities.
4. Identifying cluster centers.
5. Traversal of neighbor cells.

Additional information about grid-based clustering can be found in the following works [9, 20, 39, 45].

In sections IV and V in particular it has been described how cluster bins, derived from an m-adic tree, provide us with a grid-based framework or data structuring. We can read off the cluster bin members from such an m-adic tree. In subsection VIII A we have noted how an m-adic tree requires one scan through the data, and therefore this data structure is constructed in linear computational time.

In such a preprocessing context, clustering with the Baire distance can be seen as a “crude” method for getting clusters. After this we can use more traditional techniques to refine the clusters in terms of their membership. Alternatively (and we have quite extensively compared Baire clustering with, e.g. k-means, where it compares very well) clustering with the Baire distance can be seen as fully on a par with any optimization algorithm for clustering. As optimization, and just as one example from the many examples reviewed in this article, the Baire approach optimizes an m-adic fit of the data simply by reading the m-adic structure directly from the data.

IX. CONCLUSIONS: MANY VIEWPOINTS, VARIOUS IMPLEMENTATIONS

Baire distance is an ultrametric, so we can think of reading off observations as a tree.

Through data precision of measurement, alone, we can enhance inherent ultrametricity, or inherent hierarchical properties in the data.

Clusters in such a Baire-based hierarchy are simple “bins” and assignments are determined through a very simple hashing. (E.g. $0.3475 \rightarrow \text{bin } 3$, and $\rightarrow \text{bin } 34$, and $\rightarrow \text{bin } 347$, and $\rightarrow \text{bin } 3475$.)

Observations are mapped onto sets. (E.g. 0.3475 and 0.3462 are mapped onto sets labeled by 3 and 34 .) We have therefore a generalized ultrametric. A lattice can be used to represent range sets, leading also to a link with formal concept analysis.

Our wide range of vantage points on the Baire-based processing is important because of the many, diverse applications in view, including the structuring of the data, the reading off of clusters, the matching or best fit of observations, the determining of hierarchical properties, and so on.

Apart from showing how the Baire vantage point gives rise in practice to such a breakthrough result of having linear time hierarchical clustering our other important contribution in this work has been to show how so many vantage points can be adopted in this way on data, on the structuring and embedding of data, and ultimately on the interpretation and exploitation of data.

-
- [1] D. Achlioptas. Database-friendly random projections. In *PODS '01: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 274–281, New York, NY, USA, 2001. ACM.
 - [2] V. Anashin and A. Khrennikov. *Applied Algebraic Dynamics*. De Gruyter, 2009.
 - [3] J. Benois-Pineau, A.Y. Khrennikov, and N.V. Kotovich. Segmentation of images in p-adic and Euclidean metrics. *Dokl. Math.*, 64:450–455, 2001.
 - [4] J.P. Benzécri. *La Taxinomie*. Dunod, 2nd edition, 1979.
 - [5] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 245–250, New York, NY, USA, 2001. ACM.
 - [6] P.E. Bradley. On p-adic classification. *p-Adic Numbers, Ultrametric Analysis, and Applications*, 1:271–285, 2009.
 - [7] P.E. Bradley. Mumford dendrograms. *Journal of Classification*, 53:393–404, 2010.
 - [8] R. Buaba, A. Homaifar, M. Gebril, E. Kihn, and M. Zhizhin. Satellite image retrieval using low memory locality sensitive hashing in Euclidean space. *Earth Science Informatics*, 4:17–28, 2011.
 - [9] Jae-Woo Chang and Du-Seok Jin. A new cell-based clustering method for large, high-dimensional data in data mining applications. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 503–507, New York, NY, USA, 2002. ACM.
 - [10] P. Contreras. *Search and Retrieval in Massive Data Collections*. PhD thesis, Royal Holloway, University of London, 2010.
 - [11] P. Contreras and F. Murtagh. Fast, linear time hierarchical clustering using the Baire metric. *Journal of Classification*, 2011. In press.
 - [12] S. Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 143–151, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers.
 - [13] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.
 - [14] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2nd edition, 2002.
 - [15] S. Deegalla and H. Boström. Reducing high-dimensional data by principal component analysis vs. ran-

- dom projection for nearest neighbor classification. In *ICMLA '06: Proceedings of the 5th International Conference on Machine Learning and Applications*, pages 245–250, Washington, DC, USA, 2006. IEEE Computer Society.
- [16] D. Dutta, R. Guha, P.C. Jurs, and Ting Chen. Scalable partitioning and exploration of chemical spaces using geometric hashing. *Journal of Chemical Information and Modeling*, 46(1):321–33, 2006.
 - [17] Xiaoli Zhang Fern and C. Brodly. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference on Machine Learning*, Washington, DC, USA, 2007. AAAI Press.
 - [18] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *KDD 2003: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 517–522, New York, NY, USA, 2003. ACM.
 - [19] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.
 - [20] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics. SIAM, 2007.
 - [21] P. Grabusts and A. Borisov. Using grid-clustering methods in data classification. In *PARELEC '02: Proceedings of the International Conference on Parallel Computing in Electrical Engineering*, page 425, Washington, DC, USA, 2002. IEEE Computer Society.
 - [22] P. Hall, J.S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society B*, 67:427–444, 2005.
 - [23] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Foundations of Computer Science, FOCS 2000, Redondo Beach, CA*, pages 189–197, 2000.
 - [24] P. Indyk, A. Andoni, M. Datar, N. Immorlica, and V. Mirrokni. Locally-sensitive hashing using stable distributions. In *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006.
 - [25] M.F. Janowitz. An order theoretic model for cluster analysis. *SIAM Journal on Applied Mathematics*, 34:55–72, 1978.
 - [26] M.F. Janowitz. Cluster analysis based on posets. Technical report, 2005. lacim.uqam.ca/~sfc05/Articles/Janowitz.pdf.
 - [27] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.
 - [28] I. C. Lerman. *Classification et Analyse Ordinale des Données*. Dunod, Paris, 1981.
 - [29] P. Li, T. Hastie, and K. Church. Very sparse random projections. In *KDD 2006: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 1, pages 287–296, New York, NY, USA, 2006. ACM.
 - [30] Ping Li, T.J. Hastie, and K.W. Church. Very sparse random projections. In *KDD'06 Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages

- 287–296. ACM, New York, 2006.
- [31] J. Lin and D. Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *3rd SIAM International Conference on Data Mining*, San Francisco, CA, USA, March 2003.
 - [32] M.L. Miller, M. Acevedo Rodriguez, and I.J. Cox. Audio fingerprinting: Nearest neighbor search in high dimensional binary spaces. *Journal of VLSI Signal Processing Systems*, 41(3):285–291, 2005.
 - [33] B. Mirkin and P. Fishburn. *Group Choice*. V. H. Winston, 1979.
 - [34] F. Murtagh. Expected time complexity results for hierarchic clustering algorithms that use cluster centers. *Information Processing Letters*, 16:237–241, 1983.
 - [35] F. Murtagh. *Multidimensional Clustering Algorithms*. Physica-Verlag, 1985.
 - [36] F. Murtagh. On ultrametricity, data coding, and computation. *Journal of Classification*, 21:167–184, 2004.
 - [37] F. Murtagh. *Correspondence Analysis and Data Coding with Java and R*. Clapman & Hall/CRC, 2005.
 - [38] F. Murtagh, G. Downs, and P. Contreras. Hierarchical clustering of massive, high dimensional data sets by exploiting ultrametric embedding. *SIAM Journal on Scientific Computing*, 30(2):707–730, February 2008.
 - [39] Nam Hun Park and Won Suk Lee. Statistical grid-based clustering over data streams. *SIGMOD Record*, 33(1):32–37, 2004.
 - [40] R. Rammal, G. Toulouse, and M. A. Virasoro. Ultrametricity for physicists. *Reviews of Modern Physics*, 58(3):765–788, July 1986.
 - [41] A.K. Seda and P. Hitzler. Generalized distance functions in the theory of computation. *Computer Journal*, 53:443–464, 2010.
 - [42] A.C.M. van Rooij. *Non-Archimedean Functional Analysis*. Marcel Dekker, 1978.
 - [43] S.S. Vempala. *The Random Projection Method. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, volume 65. American Mathematical Society, 2004.
 - [44] Pengcheng Wu, S.C.H. Hoi, Nguyen Duc Dung, and He Ying. Randomly projected kd-trees with distance metric learning for image retrieval. In *International Conference on Multimedia Modeling (MMM2011)*. Taipei, Taiwan, 2011.
 - [45] Rui Xu and D.C. Wunsch. *Clustering*. IEEE Computer Society Press, 2008.